MASTER'S THESIS

Master of Science in Advanced Mathematics and Mathematical Engineering

Title: Incorporating curvature to the boundary of linear and high-order meshes when a target geometry is unavailable

Author: Albert Jiménez Ramos

Advisors: Abel Gargallo-Peiró, Xevi Roca

Research center: Barcelona Supercomputing Center

Academic year: 2017-2018

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Master in Advanced Mathematics and Mathematical Engineering

Master's thesis

Incorporating curvature to the boundary of linear and high-order meshes when a target geometry is unavailable

Albert Jiménez Ramos

Supervised by
Abel Gargallo-Peiró
Xevi Roca

October, 2018

En primer lloc, i amb una importància capital, vull agrair als meus magnífics tutors el temps que m'han dedicat. Heu estat uns tutors excepcionals. Xevi i Abel, moltes gràcies, de veritat.

Vull recordar també els meus mestres i professors. M'han format com a persona i m'han fet despertar l'interès pels mètodes numèrics primer, i per la generació de malles després. Us estic molt agraït.

Una petita part d'aquest projecte també és dels meus amics i companys de feina. M'heu divertit i m'heu ajudat a no defallir.

Per últim, vull donar-li les gràcies als meus pares, a la meva germana i a la meva família. Gràcies pel vostre esforç, la vostra paciència i el vostre suport. Des de sempre i per sempre, gràcies.

Abstract

In this work, two new methods are presented: first, a given linear tetrahedral volume mesh is refined in such a way that its boundary tends to an almost everywhere $\mathcal{C}^2$-continuous surface, and second, a curved high-order tetrahedral mesh which boundary approximates an almost everywhere $\mathcal{C}^2$ surface is generated from a linear tetrahedral mesh.

On the one hand, we propose a new subdivision method for linear tetrahedral meshes when no additional information about the boundary is provided. The boundary of the resulting finer tetrahedral mesh tends to an almost everywhere $\mathcal{C}^2$-continuous surface. We devise the method as follows. First, we subdivide each element into eight smaller tetrahedra. Second, the boundary of the initial mesh is, in particular, a triangular surface mesh, and is subdivided using the Loop subdivision scheme. Finally, the boundary of the refined volume mesh is replaced by the finer surface mesh.

On the other hand, we present a method to generate a high-order tetrahedral mesh from a linear tetrahedral mesh. For polynomial degree four, around regular points the boundary of the high-order mesh interpolates a $\mathcal{C}^2$-continuous surface; around extraordinary points this surface is approximated with fifth-order accuracy. We highlight that this method does not require an underlying curved geometry representation, only a linear tetrahedral mesh is used as input data. The method starts generating a straight-sided high-order mesh by increasing the polynomial degree of the physical elements. Next, from the boundary of the original mesh, a linear triangular surface mesh, we generate a high-order surface mesh. Finally, the boundary of the straight-sided tetrahedral mesh is replaced by the high-order surface mesh.

The aim of these curved meshes is to provide an almost everywhere $\mathcal{C}^2$ smooth parametrization of curved boundaries suitable for finite element analysis with unstructured high-order methods. To illustrate the applicability of the methods, several curved high-order volume meshes have been generated from linear meshes obtained by digitizing real volumes.

Keywords

High-order mesh generation; subdivision volume; curved meshing

Contents

Chapter 1

Introduction

1.1 Motivation

Numerical simulations of physical models described by partial differential equations (PDE) are widely used for design, dimensioning and validation purposes. One of the most used methods is the finite elements method (FEM). In this method, a continuous problem (the initial PDE model) is replaced by a discrete problem that can actually be computed and which solution is an approximate solution to the initial problem.

Therefore, the first step of such simulation consists in the discretization of the physical domain. Replacing the continuous region by means of a finite union of (geometrically simple and bounded) *elements* such as triangles, quadrilaterals, tetrahedra, hexahedra, etc., generates the computational mesh. Depending on the space dimension and the type of problem, one kind of element or another is preferred [Frey and George, 2000]. Simplicial meshes present, in general, higher geometrical flexibility. Thus, this thesis is devoted to triangles (2D) and tetrahedra (3D).

Together with the element type, different orders of accuracy of a given element can be chosen. In *linear meshes*, element entities are connected by linear polynomials. In *high-order meshes*, higher degree polynomials are used and the physical model is represented more accurately. However, the use of high-order meshes has been motivated not only because they allow us to improve the geometry accuracy, but also because it has been shown that, under some circumstances regarding the exact solution of a PDE, the approximated solution converges exponentially with the order of the approximating polynomial when one uses high-order methods [Babuška et al., 1981, Szabó and Babuška, 1991].

High-order mesh generation has focused the attention of mesh generation community in the last decade. In most cases, the domain to be discretized is given by a Computed-Aided Design (CAD) model and existing curving methods strongly use this fact. Despite that, sometimes CAD-based modeling is not feasible or desirable. In some cases, simulations systems have to deal with legacy CAD models that are somehow inconsistent when imported into current CAD systems and modern finite element solvers may fail. In other situations, such as digital terrain

modeling, CAD geometries are simply not available. In this work, the proposed method only requires a linear mesh and an underlying curved geometry representation is not needed. The aim of these curved meshes is to provide an almost everywhere $\mathcal{C}^2$ smooth parametrization of curved boundaries suitable for finite element analysis with unstructured high-order methods.

1.2 State of the art

1.2.1 Subdivision surfaces

The subdivision process for surface meshes has been extensively studied and many methods have been proposed. There are three main characteristics that differentiate the existing methods. First, a method is devised to subdivide a unique type of element. That is, a subdivision method can only be applied to refine either quadrilateral, triangular, hexahedral or tetrahedral meshes, just to name a few of them. The second important property is the method adaptiveness. Adaptive subdivision aims to provide a rule to decide whether a given element of the mesh has to be subdivided in the next step of subdivision. Finally, an important distinction regards the regularity of the surface to which the subdivision scheme tends to, known as *limit surface*.

[Catmull and Clark, 1978] proposed a non-adaptive subdivision scheme for quadrilateral meshes, with an almost everywhere $\mathcal{C}^2$-continuous limit surface. More recently, [Kobbelt, 2000] proposed an adaptive method for triangular elements that rather than inserting a new vertex for every existing edge, it adds a new vertex for every face. This method also converges to an almost everywhere $\mathcal{C}^2$ limit surface but it is hardly generalizable to three-dimensions.

The subdivision surfaces scheme presented in [Loop, 1987] refines a triangular surface mesh by generating a new node for each triangle edge. The surface meshes considered in this work correspond to the boundary of a given tetrahedral mesh. In particular, we require to couple the subdivision of the boundary mesh with the subdivision of the volume mesh. Standard volume subdivision divides all the boundary faces uniformingly in all the elements. Similarly, Loop's scheme uses a uniform subdivision of all the triangles. Hence, amongst the different surface subdivision methods, we choose to use the scheme presented by Loop.

In addition, Loop's subdivision scheme is one of the standard subdivision algorithms in the computer graphics community. One of the main features of the method is that the subdivided mesh tends to a $\mathcal{C}^2$-continuous surface when applied to regular meshes (meshes which nodes have six neighbors); while around irregular points the surface is $\mathcal{C}^1$. Interesting results have been obtained regarding the convex hull property [Loop, 2002], and more recently [Persson et al., 2006] applied Loop's subdivision for surrogate geometries.

In this thesis, we extract the boundary of a tetrahedral volume mesh and then apply Loop subdivision surfaces scheme. Next, this finer triangular surface mesh is coupled with the subdivided tetrahedral volume mesh to finally generate a finer tetrahedral volume mesh with a boundary tending to an almost everywhere $\mathcal{C}^2$-continuous surface.

1.2.2 Subdivision volume

As in the two-dimensional case, subdivision of volume meshes has been also widely studied in the last two decades. On the one hand, subdivision volume is used for volume visualization within the computer graphics community. In [Greiner and Grosso, 2000], a hierarchy of adaptive 3D meshes is generated in which finer elements are entirely covered by its parent element. Later, [Bajaj et al., 2002] proposed a subdivision method for hexahedral meshes where every new point is generated by means of an average of the neighboring points. Although not proved, the scheme seems to tend to a smooth limit surface.

On the other hand, as opposed to computer graphics where the main meshing target is the visual realism of the mesh, computational meshes requires of well defined discretizations composed of well-shaped elements. However, an inherent problem of subdividing a mesh is to recover the geometry in the new nodes on the boundaries, and, in addition, obtain a valid final mesh. To overcome this drawback several strategies have been developed. In [Yilmaz and Aliabadi, 2011, Yilmaz and Aliabadi, 2013], the new mid-nodes are relocated to the closest node in a fine surface mesh, at the same time the location of interior nodes is also modified. Similarly, in [Antepara et al., 2015] the new geometry nodes are projected to an STL representation of the geometry. In hexahedral domains that can be decomposed in structured or sweeping blocks, [Staten et al., 2016] presented a method to increase the number of elements by any positive integer (and not only powers of 2).

In this work, we present a method to refine a linear tetrahedral volume mesh. The boundary is refined accounting to approximate the almost everywhere $\mathcal{C}^2$ limit surface from the Loop subdivision scheme, see Subsection 1.2.1. The final result is a tetrahedral volume mesh which boundary tends to an almost everywhere $\mathcal{C}^2$ surface. We highlight that our method uses exclusively a linear tetrahedral volume mesh as input data.

1.2.3 High-order mesh generation

The generation of high-order meshes has been motivated by the importance of high-order methods among computational methods community. Most of the existing work relies on the information given from Computer-Aided-Design (CAD) geometries. Some works suggest to map the boundary nodes onto the CAD surface and then solve a solid body deformation problem for the interior nodes. For instance, [Xie et al., 2013] generated a curved mesh by using a linear elasticity analogy which employs boundary loads that ensure the nodes representing curved boundaries lie on the true surface, [Moxey et al., 2016] proposed a linear elasticity approach having control over the validity of the elements in the resulting mesh, while [Persson and Peraire, 2008] used non-linear elasticity prescribing a boundary displacement to be that of the curved geometry. Other techniques have been presented by [Gargallo-Peiró, 2014], [Gargallo-Peiró et al., 2015a] and [Gargallo-Peiró et al., 2015b]: the initial linear mesh is modified introducing high-order nodes, following the boundary nodes are displaced to ensure they are on the CAD surface and the interior nodes are relocated to optimize the mesh quality. In all these cases, the target smoothness is a $\mathcal{C}^0$ surface.

High-order methods rely on a high-quality curved high-order mesh that approximates the physical domain. To this end, in [Ruiz-Gironés et al., 2016b] a method is proposed to generate high-order meshes featuring optimal mesh quality and geometric accuracy. Similarly, [Ruiz-Gironés et al., 2016a] presented a curving method that converts a straight-sided mesh to an optimal-quality curved high-order mesh that interpolates the CAD boundary representation. In order to reduce the computational cost, accurate bounds on the distortion are necessary [Johnen et al., 2013].

Alternatively, in [Chaurasia et al., 2012] subdivision of a first generated coarse high-order mesh is applied to the mesh curving of moving domains. A posteriori approaches to insert boundary layers in curved meshes can also be found: in [Gargallo-Peiró et al., 2013], first the isotropic mesh is generated, and then the boundary elements are inserted; while in [Moxey et al., 2015a, Moxey et al., 2015b] boundary layers are generated by subdividing boundary prismatic elements into anisotropic tetrahedra. More recently, another method can be found in [Gargallo-Peiró et al., 2017], where a coarse linear mesh is curved to match the boundaries of the computational domain, and following the linear mesh is subdivided and then mapped to high-order mesh, resulting in a finer linear mesh with improved geometric accuracy.

However, as in tomography sectioning or topography studies, a continuous CAD model does not exist and only a linear mesh is available. [Jiao and Wang, 2012] proposed two curving methods based in weighted least squares approximations and piecewise polynomial fittings which guarantee $\mathcal{C}^0$ continuity. More recently, [Borja, 2017] used the Loop subdivision surfaces scheme to generate $\mathcal{C}^2$ almost everywhere high-order surface meshes.

In this thesis, we present a method to generate a curved high-order tetrahedral volume mesh when only a linear tetrahedral mesh is available. The boundary of the resulting mesh is a high-order triangular surface mesh. For polynomial degree four, around regular points the surface mesh interpolates an almost everywhere $\mathcal{C}^2$-continuous surface; while around extraordinary points the surface mesh approximates it with fifth-order accuracy.

1.3 Goals of the thesis and contributions

Although subdivision and high-order mesh generation have been extensively studied in the last decade, most of the available methods need the exact representation of the discretized object, either by CAD geometries, STL representations or a finer mesh. When the exact representation is unknown, less methods have been studied.

The main goal of this thesis is to address this issue by proposing two methods:

- Given a linear tetrahedral mesh, a finer linear tetrahedral mesh is generated which boundary tends to an almost everywhere $\mathcal{C}^2$-continuous surface.
- Given a linear tetrahedral mesh, a curved high-order tetrahedral mesh is generated which boundary is an almost everywhere $\mathcal{C}^2$-continuous high-order triangular surface mesh.

The rest of this thesis is organized as follows.

In Chapter 2, we introduce all the background required to understand the scheme we present later. In particular, in Section 2.1 we explain the concept of *master element*, and we carefully define the position of the nodes for any polynomial degree. In Section 2.2, we present the mapping to relate the master element with any physical element.

Chapter 3 has two distinct sections. Section 3.1 focuses to the Loop subdivision surfaces scheme. Besides that, the regularity of the limit surface is also studied. Section 3.2 explains how Loop's scheme is used to generate a curved high-order triangular surface mesh from a linear one.

In Chapter 4, the two contributions of this work are presented. In Section 4.1, we present a method to subdivide a tetrahedral volume mesh. The boundary is refined accounting to approximate the almost everywhere $\mathcal{C}^2$-continuous limit surface from the Loop subdivision scheme. Section 4.2 is devoted to the main contribution of this thesis: we generate a curved high-order tetrahedral volume mesh from a linear one. For polynomial degree 4, around regular points, the boundary of the resulting mesh interpolates an almost everywhere $\mathcal{C}^2$-continuous surface; while around extraordinary points the surface mesh approximates it with fifth-order accuracy.

Chapter 2

Preliminaries

The finite element method for the solution of PDEs requires a partition of the computational domain in simpler geometric entities, called the elements [Quarteroni and Quarteroni, 2009]. In particular, we are interested in *conformal meshes*: meshes in which the intersection between two elements is either empty or a whole common node, edge, or face. We focus on simplicial meshes: triangles in two-dimensions and tetrahedra in three-dimensions.

Although we only consider one type of element, we do not expect to have elements with the same shape. Thus, computations on this type of meshes are cumbersome. This problem can be circumvented introducing what is known as reference or *master element*. The master element is an ideal fixed element where we define a set of quantities so that calculations for a typical element may be standardized [Hughes, 1987]. Specifically, the master element is described by its domain of definition, the position and labeling of their nodes and the shape functions associated to them, see Section 2.1. On the other hand, we have the real element of the mesh living in the physical space, called physical element. We have as many physical elements as elements the mesh has, but there is only one master element. In our case, although the physical elements can be curved, the reference element always has straight-sided edges. Master and physical elements are related by the master or representation mapping, see Section 2.2.

2.1 The master element

In two dimensions, we consider triangular meshes. As a consequence, we define the two-dimensional master element as a triangle with vertices at points $(-1,-1)$, $(+1,-1)$ and $(-1,+1)$. In general, a triangular element with shape functions of polynomial degree q has $n_q = \frac{(q+1)(q+2)}{2}$ nodes, so for linear elements, $q = 1$, the three vertices uniquely define the linear triangle, see Figure 2.1(a). For $q = 2$, we need three more nodes placed at the midpoint of the edges, see Figure 2.1(b). More generally, we generate an evenly distributed grid inside the square $[-1,1]^2$ with q^2 cells, and the triangle below the top-left$-$bottom-right diagonal defines the position for the high-order triangle nodes, as illustrated in Figure 2.1(c) for polynomial degree $q = 4$.

We highlight that nodes are ordered from left to right and from bottom to top and vertices

are always on the same position, independently of the polynomial degree we are considering.

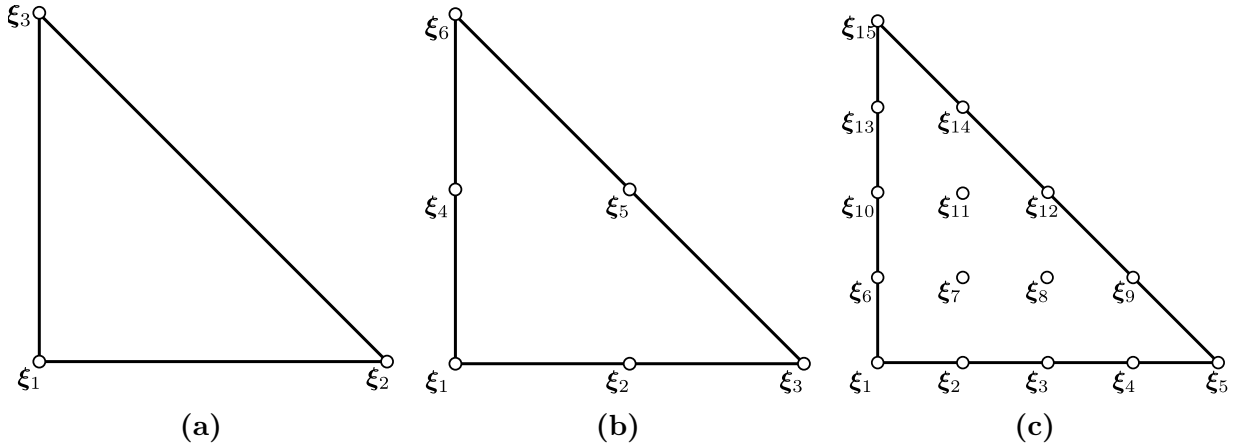


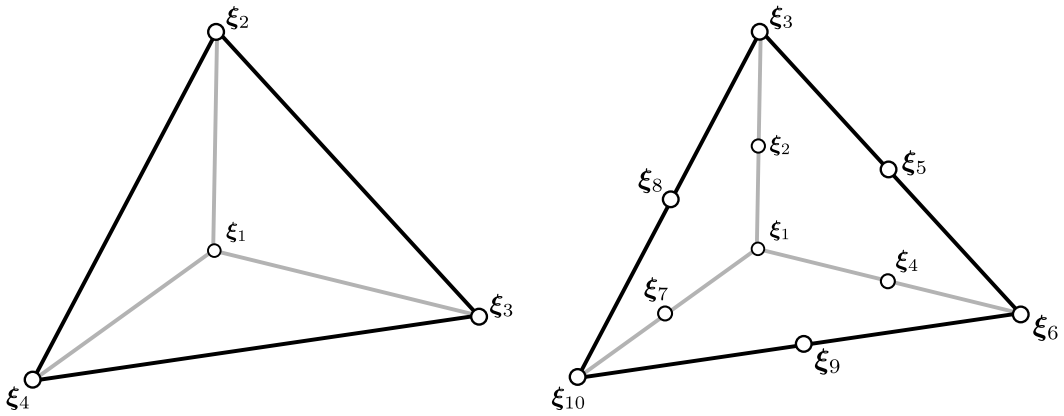
Figure 2.1: Master triangle of polynomial degree **(a)** $q = 1$, **(b)** $q = 2$ and **(c)** $q = 4$.

In three dimensions, we consider tetrahedral meshes. Hence, the master element is a tetrahedron with vertices at points $(-1, -1, -1)$, $(-1, -1, +1)$, $(-1, +1, -1)$ and $(+1, -1, -1)$. In general, a tetrahedron of polynomial degree q has $n_q = \frac{(q+1)(q+2)(q+3)}{6}$ nodes. Analogously, we generate a structured grid inside the cube $[-1, 1]^3$ and cut by the plane passing through three of the vertices. The four vertices define a linear tetrahedron, see Figure 2.2(a), while ten nodes uniquely define the reference element of polynomial degree $q = 2$, see Figure 2.2(b). For fourth polynomial degree tetrahedron, as well as for higher-order elements, there are nodes on the faces and inside the tetrahedron.

Note that the nodes follow a particular order. For instance, for polynomial degree $q = 2$, the nodes are at positions

$$\begin{array}{ll} \boldsymbol{\xi}_1 = (-1,\ -1,\ -1) & \boldsymbol{\xi}_6 = (-1,\ 1,\ -1) \\ \boldsymbol{\xi}_2 = (-1,\ -1,\ 0) & \boldsymbol{\xi}_7 = (0,\ -1,\ -1) \\ \boldsymbol{\xi}_3 = (-1,\ -1,\ 1) & \boldsymbol{\xi}_8 = (0,\ -1,\ 0) \\ \boldsymbol{\xi}_4 = (-1,\ 0,\ -1) & \boldsymbol{\xi}_9 = (0,\ 0,\ -1) \\ \boldsymbol{\xi}_5 = (-1,\ 0,\ 0) & \boldsymbol{\xi}_{10} = (1,\ -1,\ -1) \end{array}$$

We highlight that nodes 1, $q + 1$, $\frac{(q+1)(q+2)}{2}$ and $\frac{(q+1)(q+2)(q+3)}{6}$ always correspond to the vertices of the tetrahedron of polynomial degree q.



(a) **(b)**

Figure 2.2: Master tetrahedron of polynomial degree **(a)** $q = 1$ and **(b)** $q = 2$.

2.2 From the master to the physical element

As we have already mentioned, a unique master element may serve to characterize all the elements in the mesh. Moreover, a part of the computations for a given element concerns the master element and therefore it does not change when we consider one element or another of the mesh. Thus, it is crucial to relate both domains. In the linear case, there exists a unique affine mapping, $\boldsymbol{\phi}$, from the master element to the physical one. In the high-order case, the mapping $\boldsymbol{\phi}$ is no longer affine. In any case, every point $\boldsymbol{x}$ on the physical element is the image by $\boldsymbol{\phi}$ of a point $\boldsymbol{\xi}$ on the reference element.

Consider that the physical element is a nodal high-order simplex of degree q determined by n_q nodes with coordinates $\boldsymbol{x}_i$, for $i = 1, ..., n_q$. Hence, the master element is characterized by a node distribution $\boldsymbol{\xi}_i$, for $i = 1, ..., n_q$. Therefore, the following holds

$$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{\xi}) = \sum_{i=1}^{n_q} \boldsymbol{x}_i N_i(\boldsymbol{\xi}) \tag{2.1}$$

where $\{N_i\}_{i=1,...,n_q}$ is a basis of nodal shape functions. For a full discussion about these basis functions and their properties see [Hesthaven and Warburton, 2007]. Furthermore, we have $\boldsymbol{x}_i = \boldsymbol{\phi}(\boldsymbol{\xi}_i)$, for $i = 1, ..., n_q$.

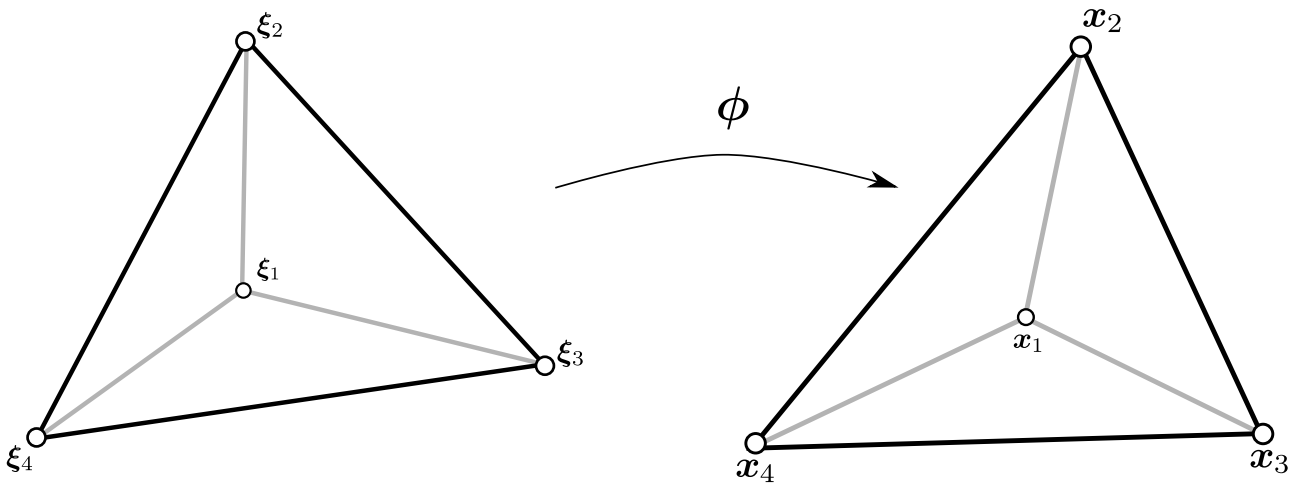


Figure 2.3: Mapping from the master element onto the physical element.

This mapping is extremely useful when solving a PDE with, for instance, the finite element method. A linear system is created by assembling the contributions of each mesh element to the system matrix. These contributions can be computed by integrating directly in the physical element or by changing the variable and integrating in the master element. To this end, $\boldsymbol{\phi}$ is required to be a differentiable, invertible and smooth mapping (diffeomorphism). Hence, the mapping has to be expressed by means of differentiable functions and the mesh elements have to be valid (non-folded) and present high-quality (regular shape). If one element is invalid then the determinant of the mapping Jacobian presents non-positive values that invalidate the change of variable, and therefore, the obtained solution. Moreover, if one element has low quality, then the element is distorted respect a regular element. In summary, quality measures have to be used to assess the validity and quality of a given mesh [Gargallo-Peiró, 2014].

Chapter 3

Surface mesh: from straight-sided to curved

This chapter is devoted to triangular surface meshes. In Section 3.1, we explain how we refine a linear triangular surface mesh by means of the Loop subdivision scheme [Loop, 1987]. In Section 3.2, given a linear surface mesh composed of straight-sided triangles, we generate a curved high-order surface mesh with triangles of polynomial degree two and four. For polynomial degree four, the mesh is $\mathcal{C}^2$-continuous almost everywhere and interpolates the limit surface.

3.1 Subdivision surfaces

In this section, we present the Loop subdivision scheme. It is composed of two main steps: first, new nodes for each edge have to be generated and second, the connectivity for the finer mesh has to be defined. In particular, each element is divided into four smaller elements. Formally, given a linear mesh $\mathcal{M}^0$ with n^0_{nodes} nodes and n^0_{elem} elements, we obtain a new linear mesh $\mathcal{M}^1$ with $n^1_{\text{nodes}} = n^0_{\text{edges}} + n^0_{\text{nodes}}$ nodes and $n^1_{\text{elem}} = 4 \cdot n^0_{\text{elem}}$ elements, where n^0_{edges} is the number of edges of $\mathcal{M}^0$. The rest of this section is structured as follows: in Subsection 3.1.1 we detail how new nodes are generated, while in Subsection 3.1.2 we proceed to the actual refinement of the mesh. In Subsection 3.1.3, we discuss to which surface this subdivision scheme tends to, known as the *limit surface*, and its regularity.

3.1.1 New nodes position

In the Loop subdivision algorithm a new node is generated for each edge. Note that an edge connects two nodes and belongs to two elements. Let $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_4$ be four points in $\mathbb{R}^3$ that define two triangles $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$ and $(\boldsymbol{x}_1, \boldsymbol{x}_4, \boldsymbol{x}_2)$. These triangles share the edge $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, as shown in

Figure 3.1. In this edge, a new node $\boldsymbol{x}_{\widehat{12}}$ is generated using the expression

$$\boldsymbol{x}_{\widehat{12}} = \frac{3}{8}\left(\boldsymbol{x}_1 + \boldsymbol{x}_2\right) + \frac{1}{8}\left(\boldsymbol{x}_3 + \boldsymbol{x}_4\right) \tag{3.1}$$

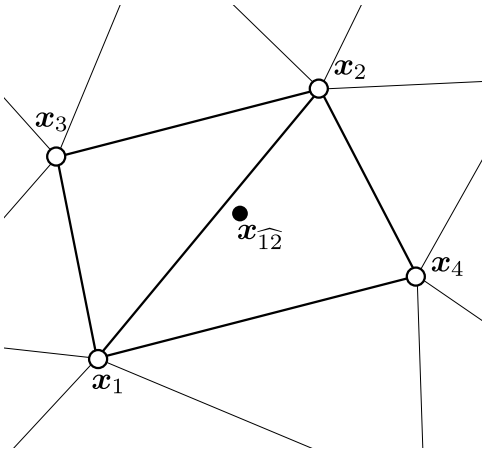


Figure 3.1: A new node is generated on the edge shared by triangles (1, 2, 3) and (1, 4, 2), as explained in Equation (3.1).

However, together with generating the new edge nodes, Loop's scheme also modifies the location of the vertices of the initial mesh. To this end, each node $\boldsymbol{x}_M$ is given a weight, ω, as a function of its number of neighbors, k, defined by

$$\omega = \frac{1}{k}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\left(\frac{2\pi}{k}\right)\right)^2\right)$$

Thus, at refinement level l, $l = 0, 1, ...$, the position of an existing node is modified according to

$$\boldsymbol{x}_M^{l+1} = \left(1 - k\omega\right)\boldsymbol{x}_M^l + \omega\sum_{i=1}^{k}\boldsymbol{x}_i^l \tag{3.2}$$

where $\boldsymbol{x}_j^l$ denotes the position of node j at refinement level l, and $\{\boldsymbol{x}_i^l\}_{i=1,...,k}$ are the neighbors of $\boldsymbol{x}_M^l$. Note that as k increases, ω decreases and $(1 - k\omega)$ increases and thus, if the node has a large number of neighbors k, the modified position $\boldsymbol{x}_M^{l+1}$ stays closer to the original position than in the case with less neighbors. In [Loop, 1987] we can find a detailed discussion on how weight coefficients, called *masks*, are obtained.

Note that since we generate a new node for each edge, the new mesh has $n^0_{\text{edges}} + n^0_{\text{nodes}}$ nodes. We highlight that the position of the nodes that form the new finer linear mesh is different from those from the original mesh.

3.1.2 New connectivity

Consider the original element e, as in Figure 3.2(a), defined by the nodes 1, 2 and 3 with coordinates $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ as

$$e = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$$

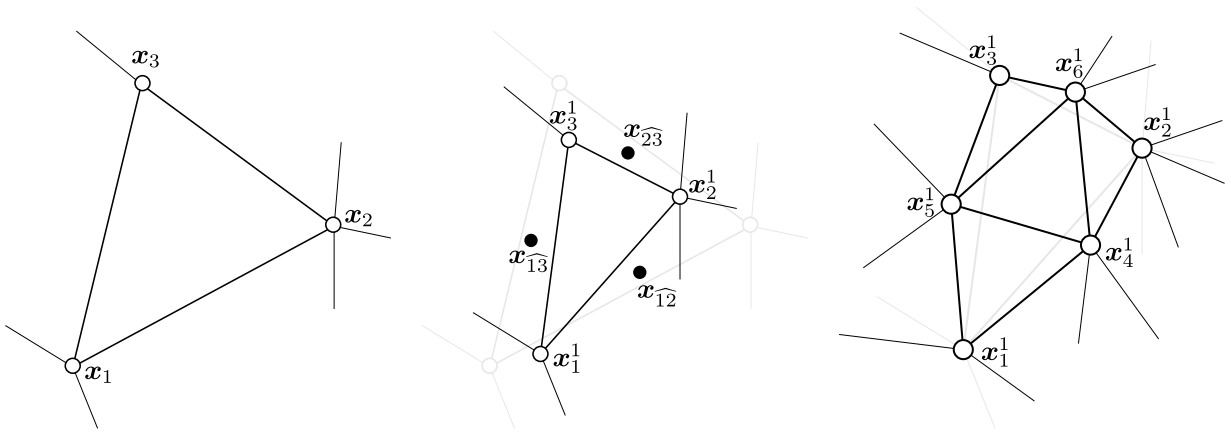
Following Subsection 3.1.1, we generate the new nodes and modify the position of the old ones, as shown in Figure 3.2(b). We denote by $\widehat{ab}$ the node with coordinate $\boldsymbol{x}_{\widehat{ab}}$ generated on the edge connecting nodes $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. If we create three edges between the three new generated nodes, each original element is divided into four smaller triangles. Specifically, the subdivision process generates the following elements

$$\begin{aligned} \widehat{e}_1 &= \begin{pmatrix} 1 & \widehat{12} & \widehat{13} \end{pmatrix} \\ \widehat{e}_2 &= \begin{pmatrix} 2 & \widehat{23} & \widehat{12} \end{pmatrix} \\ \widehat{e}_3 &= \begin{pmatrix} 3 & \widehat{13} & \widehat{23} \end{pmatrix} \\ \widehat{e}_4 &= \begin{pmatrix} \widehat{12} & \widehat{23} & \widehat{13} \end{pmatrix} \end{aligned}$$

At this point, the *auxiliary* nodes $\widehat{ab}$ are characterized by a global identifier in the new mesh. For illustrative purposes, let us assume that these three new nodes get the global numbering 4, 5 and 6, respectively. Therefore, as detailed in Figure 3.2(c), the elements in the new finer linear mesh are

$$\begin{aligned} e_1 &= \begin{pmatrix} 1 & 4 & 5 \end{pmatrix} \\ e_2 &= \begin{pmatrix} 2 & 6 & 4 \end{pmatrix} \\ e_3 &= \begin{pmatrix} 3 & 5 & 6 \end{pmatrix} \\ e_4 &= \begin{pmatrix} 4 & 6 & 5 \end{pmatrix} \end{aligned}$$

Repeating these steps for all the elements on the original mesh, we end up with a finer mesh with $4 \cdot n^0_{\text{elem}}$ elements and $n^0_{\text{edges}} + n^0_{\text{nodes}}$ nodes.



(a) **(b)** **(c)**

Figure 3.2: Subdivision process. **(a)** Element of the original mesh. **(b)** Original element once the new nodes (black dots) have been generated and the position of the old ones (white dots) have been modified. **(c)** Subdivided element after one iteration of the process.

3.1.3 The limit surface

The Loop subdivision scheme starts with a linear mesh, known as *control-mesh*, and produces one new refined mesh at each step that tends to a unique *limit surface* determined by the initial surface mesh. The limit surface is $\mathcal{C}^2$-continuous everywhere except at irregular (or extraordinary) nodes, that is, nodes that do not have six neighbors. Around irregular nodes it still retains $\mathcal{C}^1$ continuity [Loop, 1987].

Note that edges connecting extraordinary points generate ordinary points. This is so since an edge node is adjacent to, apart from the two nodes defining the edge, two more nodes by each one of the two adjacent elements to the edge. Therefore, since the percentage of regular nodes increases with each iteration, the discontinuity around irregular nodes can be mitigated applying successively the scheme.

We remark that the subdivision scheme defines a hierarchy of control-meshes, all of them converging to the same limit surface. Moreover, we can compute the limiting location for the nodes at any refinement level. In particular, given a linear mesh, the limit position for the node M at any refinement level, denoted by $\boldsymbol{x}_M^\infty$, is

$$\boldsymbol{x}_M^\infty = (1 - k\chi)\, \boldsymbol{x}_M + \chi \sum_{i=1}^{k} \boldsymbol{x}_i \tag{3.3}$$

where the weights are computed as

$$\chi = \frac{1}{k + \frac{3\omega}{8}}$$

We highlight that there exists a parametrization of this surface around regular points. This parametrization is indeed a piecewise polynomial of degree four, and its explicit expression can

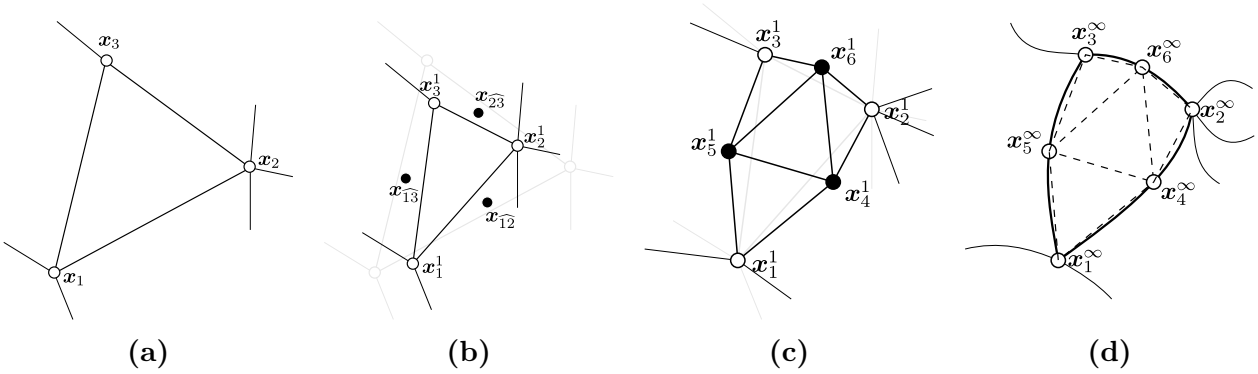
be found in [Persson et al., 2006]. The evaluation around irregular nodes is reduced to applying enough times the Loop subdivision surface process, since it introduces new regular nodes at each iteration [Persson et al., 2006]. An alternative method based on the eigenstructure of some matrices is given in [Stam, 1998].

3.2 From straight-sided to high-order curved elements

In this section, we present the surface mesh curving procedure based on subdivision that we propose in this work. Herein, given a linear surface mesh, we increase the degree of the elements and we use the subdivision surface method to relocate the high-order nodes. In this manner, the high-order nodes are located to retrieve an almost everywhere $\mathcal{C}^2$ mesh for $q \geq 4$.

Specifically, given an element of the linear mesh, Figure 3.3(a), one step of the subdivision scheme generates three edge nodes and modifies the position of the vertices, Figure 3.3(b), as explained in Subsection 3.1.1. Instead of connecting the edge nodes and subdividing the element as in Figure 3.3(c), the nodes are mapped onto its limiting position using Equation (3.3). Finally, a curved element of polynomial degree two is obtained by considering the new generated nodes as the mid-edge nodes that are required to define an element of polynomial degree two. The reinterpretation of the topology of the four subdivided elements into a unique element of degree two is illustrated in Figure 3.3(d). Following the notation in Figure 3.3, the linear element $e_{lin} = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$ generates the element of polynomial degree two $e_{ho} = \begin{pmatrix} 1 & 4 & 2 & 5 & 6 & 3 \end{pmatrix}$.

This process is repeated for all the elements, obtaining a curved surface mesh of polynomial degree two that approximates the almost everywhere $\mathcal{C}^2$ limit surface with third order accuracy. This high-order mesh has the same number of nodes as in the subdivision process but it has the same number of elements as the original linear mesh.



(a) **(b)** **(c)** **(d)**

Figure 3.3: Generation of an element of polynomial degree two from a linear one. **(a)** Straight-sided element of the original linear mesh. **(b)** Original element once the new nodes (black dots) have been generated and the position of the old ones (white dots) have been modified following the rules of the Loop subdivision scheme. **(c)** Subdivision of the linear element. **(d)** Curved element of polynomial degree two, displaying with dashed lines the four elements from the subdivision scheme.

For quartic polynomial degree $q = 4$, each element is defined by $n_4 = 15$ nodes. The procedure to generate the curved mesh of degree four from a given linear mesh is similar to the quadratic case. Applying twice the subdivision process twelve new nodes are obtained, see Figure 3.4(c). Together with the three original nodes, we have a total of fifteen nodes which uniquely define an element of polynomial degree four. Finally, these nodes are mapped onto its limiting position and the curved element of polynomial degree four is defined, see Figure 3.4(d). Following the notation in Figure 3.4, the linear element $e_{lin} = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$ generates the element of polynomial degree four $e_{ho} = \begin{pmatrix} 1 & 7 & 4 & 9 & 2 & 8 & 13 & 14 & 10 & 5 & 15 & 6 & 11 & 12 & 3 \end{pmatrix}$.

This process is repeated for all the elements, obtaining a curved surface mesh of polynomial degree four that interpolates around regular points the almost everywhere $\mathcal{C}^2$ limit surface. This high-order surface mesh has the same number of nodes as after applying two times the subdivision scheme to the original linear mesh, although the number of elements does not change.

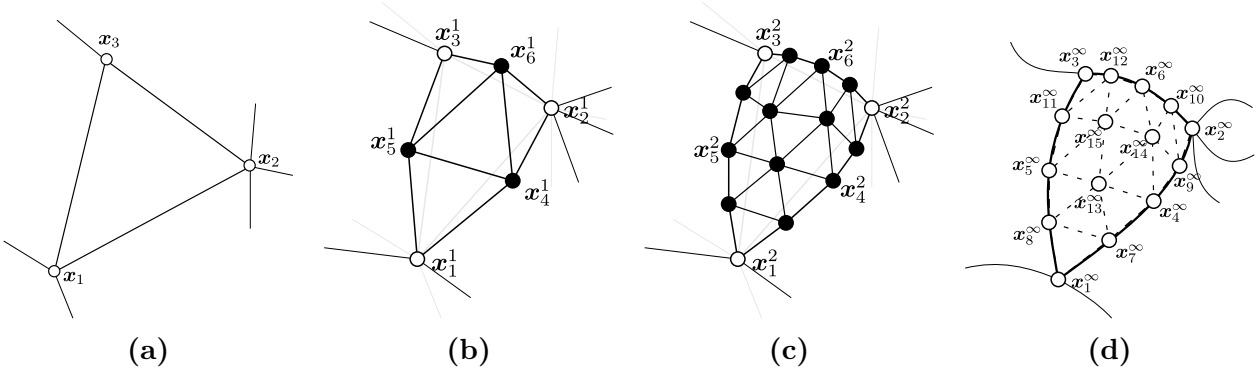


Figure 3.4: Generation of an element of polynomial degree four from a linear one. **(a)** Straight-sided element of the original linear mesh. **(b)** Subdivision of the linear element after one iteration of the subdivision surface algorithm. **(c)** Subdivision of the linear element after two iterations of the subdivision surface algorithm. **(d)** Curved element of polynomial degree four, displaying with dashed lines the sixteen elements from the subdivision scheme.

The reason why we have restricted to polynomial degree $q = 2$ and $q = 4$ is because the Loop subdivision procedure naturally gives the amount of nodes necessary to retrieve these high-order meshes. Moreover, since the limit surface is a piecewise polynomial of degree four around regular nodes, meshes of polynomial degree four capture it exactly around these points. Therefore, on the one hand, around regular nodes we are able to interpolate the limit surface and obtain a $\mathcal{C}^2$ surface mesh. On the other hand, around irregular nodes, the limit surface is approximated with fifth order accuracy.

Chapter 4

Volume mesh: from linear to curve

In this chapter, we focus on the main objectives of this thesis: the subdivision of linear tetrahedral volume meshes and the generation of high-order tetrahedral volume meshes. In Section 4.1, we detail our method to subdivide a linear tetrahedral volume mesh which boundary is refined accounting to approximate the almost everywhere $\mathcal{C}^2$ limit surface from the Loop subdivision scheme, see Subsection 3.1.3. In Section 4.2, given a linear volume mesh composed of straight-sided tetrahedra, we propose a new method to generate a curved high-order volume mesh of polynomial degrees two and four. For polynomial degree four, the boundary of this volume mesh is an almost everywhere $\mathcal{C}^2$-continuous surface mesh and interpolates the limit surface around regular nodes.

4.1 Linear tetrahedral mesh subdivision

In this section, we present a method to refine a linear tetrahedral mesh where the boundary of the resulting mesh tends to the $\mathcal{C}^2$-continuous limit surface from Loop's subdivision scheme presented in Chapter 3. Formally, given a linear mesh $\mathcal{M}^0$ with n^0_{nodes} nodes and n^0_{elem} elements, we obtain a new linear mesh $\mathcal{M}^1$ with $n^1_{\text{nodes}} = n^0_{\text{edges}} + n^0_{\text{nodes}}$ nodes and $n^1_{\text{elem}} = 8 \cdot n^0_{\text{elem}}$ elements, where n^0_{edges} is the number of edges of $\mathcal{M}^0$. Moreover, the boundary of $\mathcal{M}^1$ is the result of the Loop subdivision surfaces scheme explained in Section 3.1 applied to the boundary of $\mathcal{M}^0$.

The proposed method is composed of three main steps. First, in Subsection 4.1.1 we subdivide the straight-sided linear volume mesh into a finer straight-sided linear mesh with the same geometric realization as the original one. Each tetrahedron generates 8 smaller tetrahedra. Second, the boundary of the original volume mesh is considered separately as a surface mesh to which we apply the Loop subdivision surfaces scheme, as previously described in Section 3.1. Third, in Subsection 4.1.2 we replace the boundary of the mesh obtained in the first step by the fine surface mesh from the second step. As a result, we obtain a finer volume mesh which boundary tends to the limit surface.

4.1.1 Straight-sided tetrahedral mesh subdivision

The subdivision of a tetrahedron is a procedure analogous to the subdivision of a triangle. Roughly speaking, we place a node in the middle of each edge, obtaining ten nodes on each element that become the vertices of the smaller tetrahedra. Specifically, consider first the linear physical element $e = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$ given by the nodes with position $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ in $\mathbb{R}^3$. In this work, we consider a master element with the same vertices for any polynomial degree, see Section 2.1. In particular, the vertices position for the master element of polynomial degree two is the same as for the linear master element, although the former has more nodes. Thus, the nodes of the master element of polynomial degree two are considered as points in the linear master element, and mapped onto the physical element e by the linear master mapping. The vertices of the master element are mapped to the vertices of the physical element, and six new nodes are generated on the edges, denoted by $\boldsymbol{x}_5$, ..., $\boldsymbol{x}_{10}$ for illustrative purposes (see Figure 4.1).

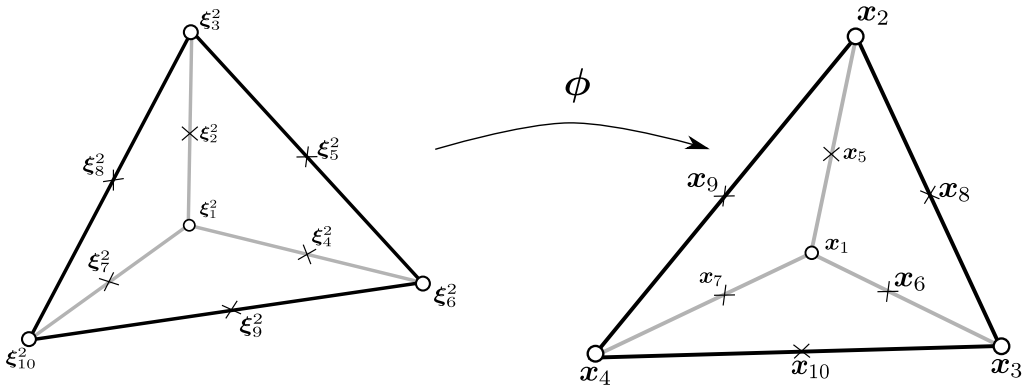


Figure 4.1: Mapping from the linear master element to the linear physical element. Ten points, corresponding to the nodes of the master element of polynomial degree two, ξ_i^2, for $i = 1, ..., 10$, are mapped from the linear master element onto the linear physical elements.

Using these ten points as vertices, we subdivide the linear element into eight smaller tetrahedra, as detailed in Figure 4.2.

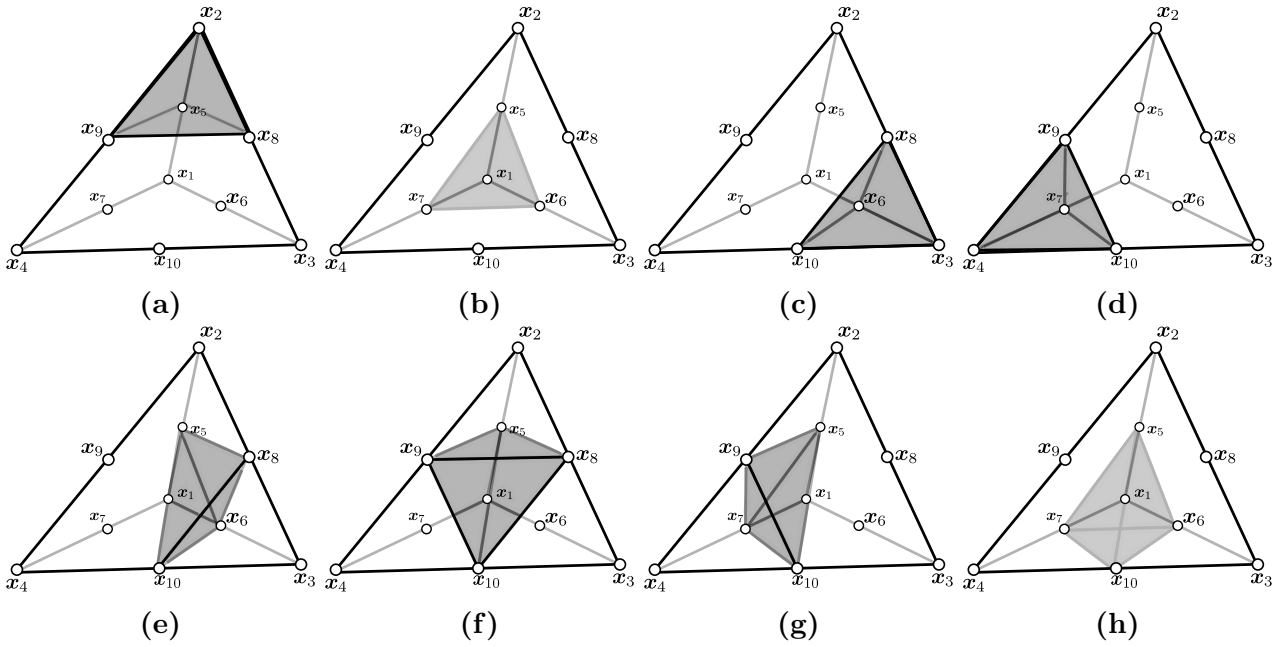


Figure 4.2: Subdivision of the master element into smaller tetrahedra. The order of the figures corresponds to elements e_i, for $i = 1, ..., 8$, in Equation (4.1).

Specifically, the obtained elements are:

$$\begin{aligned}
e_1 &= \begin{pmatrix} 9 & 5 & 8 & 2 \end{pmatrix} \\
e_2 &= \begin{pmatrix} 7 & 6 & 5 & 1 \end{pmatrix} \\
e_3 &= \begin{pmatrix} 6 & 10 & 8 & 3 \end{pmatrix} \\
e_4 &= \begin{pmatrix} 10 & 7 & 9 & 4 \end{pmatrix} \\
e_5 &= \begin{pmatrix} 6 & 10 & 5 & 8 \end{pmatrix} \\
e_6 &= \begin{pmatrix} 10 & 9 & 5 & 8 \end{pmatrix} \\
e_7 &= \begin{pmatrix} 7 & 10 & 9 & 5 \end{pmatrix} \\
e_8 &= \begin{pmatrix} 10 & 7 & 6 & 5 \end{pmatrix}
\end{aligned} \tag{4.1}$$

Once we apply this technique to all the elements of $\mathcal{M}^0$, we have an intermediate linear mesh $\widetilde{\mathcal{M}^1}$ with the same number of elements and nodes as the desired mesh $\mathcal{M}^1$, but with the same geometric boundary representation as $\mathcal{M}^0$.

4.1.2 The boundary

In this section, we focus on the boundaries of volume meshes. The boundary of a tetrahedral mesh is a triangular mesh. Thus, the method presented in Chapter 3 can be used to generate a finer surface mesh that discretizes the almost everywhere $\mathcal{C}^2$ surface given by Loop's subdivision scheme.

Let $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_4$ be four points in $\mathbb{R}^3$ that define an element of the original linear volume mesh $\mathcal{M}^0$, $e = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$. For simplicity, to detail the boundary modifications we consider that the face (2 3 4) belongs to the boundary, see Figure 4.3(a). The triangle (2 3 4) is, in particular, an element of the boundary surface mesh. Thus, on the one hand, this boundary triangle is refined by means of the Loop subdivision surfaces scheme presented in Section 3.1, as illustrated in Figure 4.3(b). Let us denote the resulting finer surface mesh by $\mathcal{S}^1$.

On the other hand, after one step of the subdivision volume algorithm explained in Subsection 4.1.1, the element e is subdivided into eight smaller tetrahedra, as shown in Figure 4.3(c). At this point, the position of boundary nodes of $\widetilde{\mathcal{M}^1}$ is replaced by the position of the equivalent nodes of the surface mesh $\mathcal{S}^1$. The resulting mesh $\mathcal{M}^1$ is a finer linear volume mesh formed by the elements and interior nodes of the refined mesh $\widetilde{\mathcal{M}^1}$ but with boundary given by $\mathcal{S}^1$. Figure 4.3(d) illustrates the resulting refined mesh for the considered case.

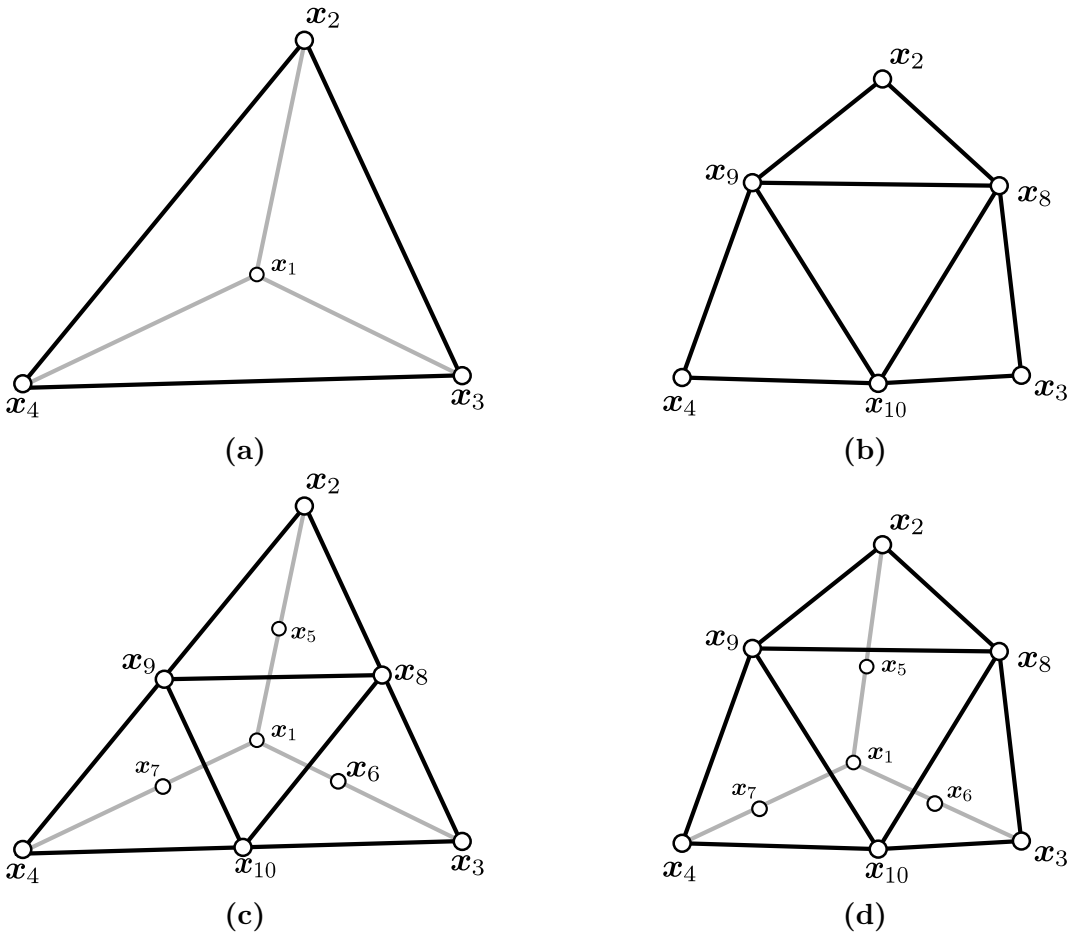
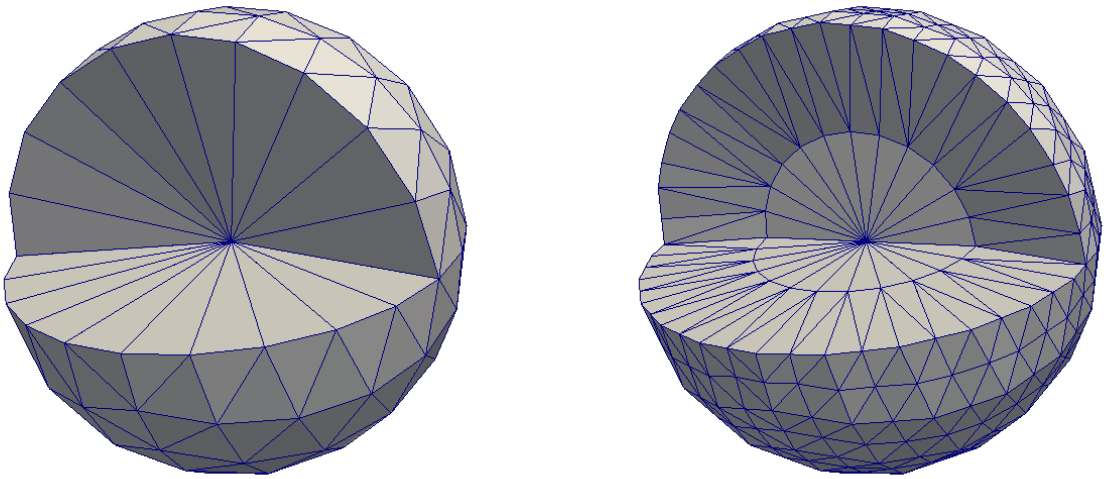


Figure 4.3: Replacement of the boundary in a linear volume element. **(a)** Original linear element $e = (1\ 2\ 3\ 4)$, where the face (2 3 4) belongs to the boundary. **(b)** Loop subdivision of the triangle face on the boundary. **(c)** Straight-sided volume subdivision of the original linear element e. **(d)** Final result of the subdivision of a boundary element. In all figures, only the subdivision of the faces of tetrahedra which belong to the boundary is illustrated.

4.1.3 A complete example: the sphere

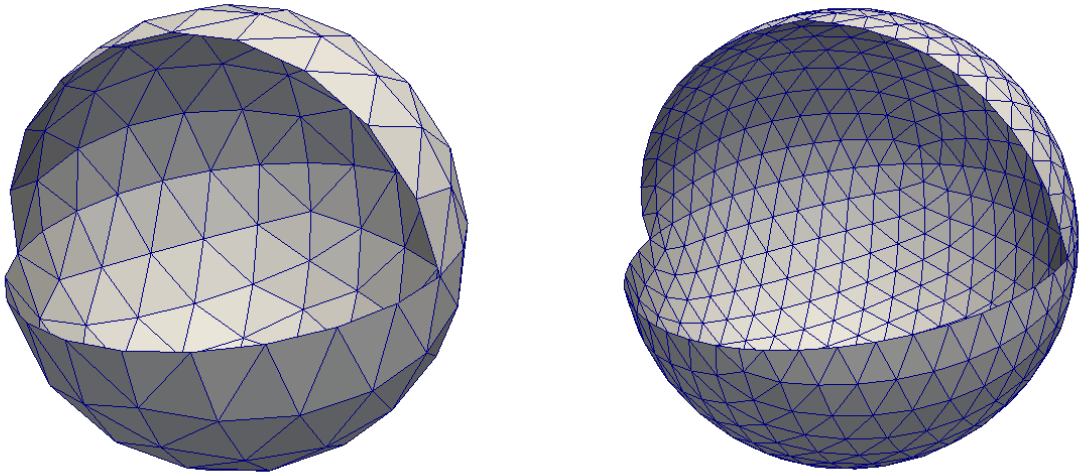
In this section we present a synthetic example to illustrate the new proposed subdivision process. Let us consider a linear volume mesh $\mathcal{M}^0$ of a sphere, as shown in Figure 4.4(a). The first step is to refine the whole volume mesh as explained in Subsection 4.1.1. The finer mesh $\widetilde{\mathcal{M}^1}$ is represented in Figure 4.4(b).



(a) **(b)**

Figure 4.4: Straight-sided subdivision of a volume mesh. **(a)** Initial linear volume mesh $\mathcal{M}^0$. **(b)** Refined volume mesh $\widetilde{\mathcal{M}^1}$.

As previously detailed, the volume mesh $\mathcal{M}^0$ is composed of tetrahedra and its boundary is a triangular surface mesh, see Figure 4.5(a). Then, we can apply one refinement of the Loop subdivision surfaces algorithm to the boundary to obtain a finer surface mesh $\mathcal{S}^1$, see Figure 4.5(b).



(a) **(b)**

Figure 4.5: **(a)** Boundary of the volume mesh $\mathcal{M}^0$. **(b)** One step of the Loop subdivision surfaces algorithm is applied and a finer surface mesh $\mathcal{S}^1$ is obtained.

Finally, we exchange the current boundary of $\widetilde{\mathcal{M}^1}$ by the finer surface mesh $\mathcal{S}^1$. The final volume mesh $\mathcal{M}^1$ is shown in Figure 4.6. The boundary tends to the almost everywhere $\mathcal{C}^2$-continuous limit surface of the Loop subdivision surfaces scheme.

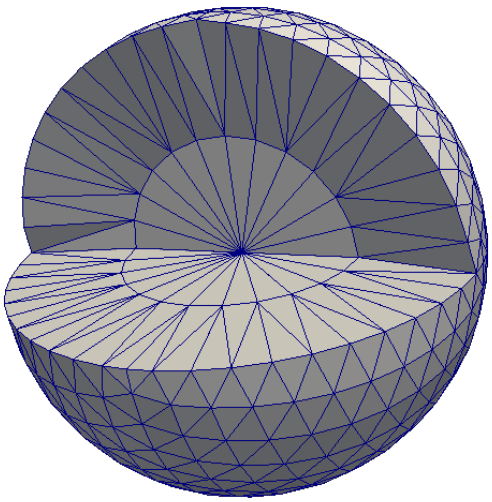


Figure 4.6: Final linear tetrahedral mesh once one refinement of the proposed volume subdivision algorithm has been applied.

4.2 From straight-sided to high-order curved elements

In this section, we present a method to generate a high-order tetrahedral mesh from a linear tetrahedral mesh. The process is very similar to the one we used in Section 4.1 to subdivide a linear volume mesh. The method is composed of three main steps. First, we generate a straight-sided high-order volume mesh from a given linear volume mesh. Next, we generate a curved high-order surface mesh from the boundary of the original linear volume mesh using Section 3.2. Then, we exchange the boundary of the straight-sided high-order volume mesh by the curved high-order surface mesh. The boundary of the final curved high-order tetrahedral mesh approximates an almost everywhere $\mathcal{C}^2$-continuous surface.

The first step is the generation of a high-order volume mesh with straight-sided edges from a linear volume mesh. Consider the physical element given be the points $\boldsymbol{x}_1$, ..., $\boldsymbol{x}_4$ in $\mathbb{R}^3$, as shown in Figure 4.7(a). The goal is to generate an element of polynomial degree q with $n_q = \frac{1}{6}(q+1)(q+2)(q+3)$ nodes. Since the master tetrahedron is the same for all orders, the n_q nodes of the high-order master element are considered as points in the linear master element, and mapped onto the physical element, see Figure 4.7(b). This straight-sided element with n_q nodes uniquely defines an element of polynomial degree q, as illustrated in Figure 4.7(c). Repeating these steps for all the elements of the original linear mesh, the straight-sided high-order volume mesh is generated. Although we have restricted to polynomial degree two and four, this method allows generating straight-sided high-order meshes of any degree.

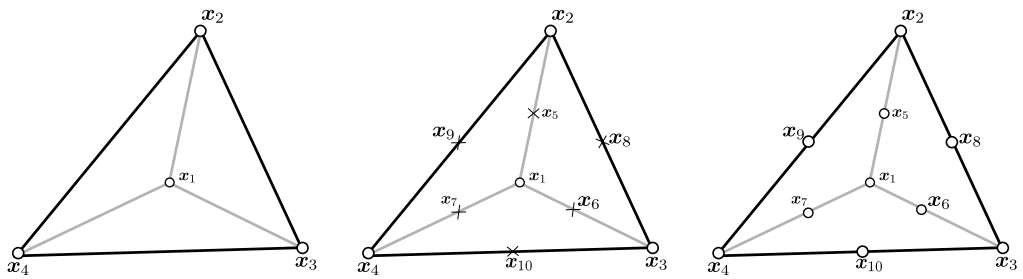


Figure 4.7: Generation of an element of polynomial degree 2 from a linear element. **(a)** Linear physical element. **(b)** Linear physical element and the image by the linear master mapping of the nodes of the master tetrahedron of polynomial degree two (crosses). **(c)** Physical element of polynomial degree two with straight-sided edges.

Secondly, the boundary of the linear volume mesh is a linear surface mesh formed by triangles. Therefore, the procedure explained in Section 3.2 can be applied to generate a curved high-order surface mesh of the same order as the desired high-order volume mesh. Third, the boundary of the straight-sided high-order volume mesh is replaced by the curved high-order surface mesh, as shown in Figure 4.8(c). This new mesh, with the same number of elements as the original linear one, is a high-order volume mesh which elements are straight-sided except those belonging to the boundary, which are curved.

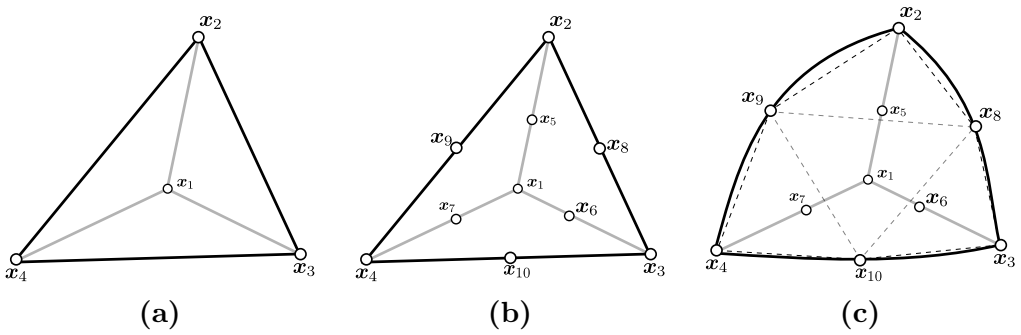


Figure 4.8: Curving of a boundary element of polynomial degree 2. **(a)** Linear physical element, where the face (2 3 4) belongs to the boundary. **(b)** Straight-sided physical element of polynomial degree two. **(c)** Curved boundary element of polynomial degree two, displaying with dashed lines the four elements from the subdivision scheme applied to the boundary.

Note that the boundary for the quadratic case, $q = 2$, determines an approximation of the almost everywhere $\mathcal{C}^2$ limit surface with third-order accuracy. On the contrary, for the quartic case, $q = 4$, the boundary exactly captures the almost everywhere $\mathcal{C}^2$-continuous limit surface around regular nodes of the original linear mesh. Around irregular nodes, the limit surface is approximated with fifth-order accuracy.

4.2.1 A complete example: the sphere

Following with the example in Subsection 4.1.3, the mesh of polynomial degree four obtained after the application of the proposed method is shown in Figure 4.9. We highlight that the boundary interpolates an almost everywhere $\mathcal{C}^2$ surface around regular points, and approximates it with fifth-order accuracy around extraordinary nodes.

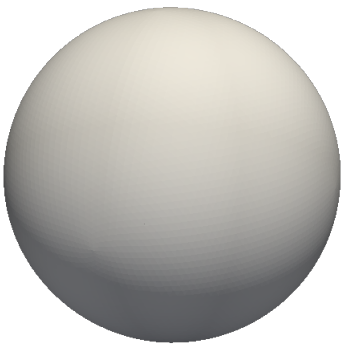


Figure 4.9: Final mesh of polynomial degree four once the proposed high-order mesh generation algorithm has been applied.

Chapter 5

Results

In this chapter, we present several examples to illustrate the main features of the proposed approach. Our sample of meshes is composed of a standard example in the computer graphics community, the Standford bunny, and four real case meshes: two of them obtained from topographic data, the Zion and Grand Canyon national parks; and two more obtained from tomography sectioning, the right hemisphere of the human brain and human rib cage.

The geometries used in this work correspond to triangle surface meshes extracted from the *GrabCAD* repository [GrabCAD, 2009], and for the bunny, from *The Standford 3D Scanning Repository* [Stanford University, 1993]. Given the geometry mesh, *MeshLab*, an open source system for processing and editing 3D triangular meshes [Cignoni et al., 2008], has been used to ensure water-tight (the mesh properly connects to adjacent surfaces around the perimeter so that the volume is fully enclosed) and conformal surface meshes. Once a valid geometry surface mesh has been obtained, the Delaunay tetrahedral mesh generator *TetGen* [Si, 2015] has been used to generate the initial input meshes for our mesh curving code.

The subdivision and mesh curving procedures have been implemented in this thesis using *Python* [Van Rossum and Drake Jr, 1995]. In particular, we used the *SciPy* ecosystem [Jones et al., 2001] and *modepy* repository [Klöckner et al., 2013] for the shape functions for high-order elements.

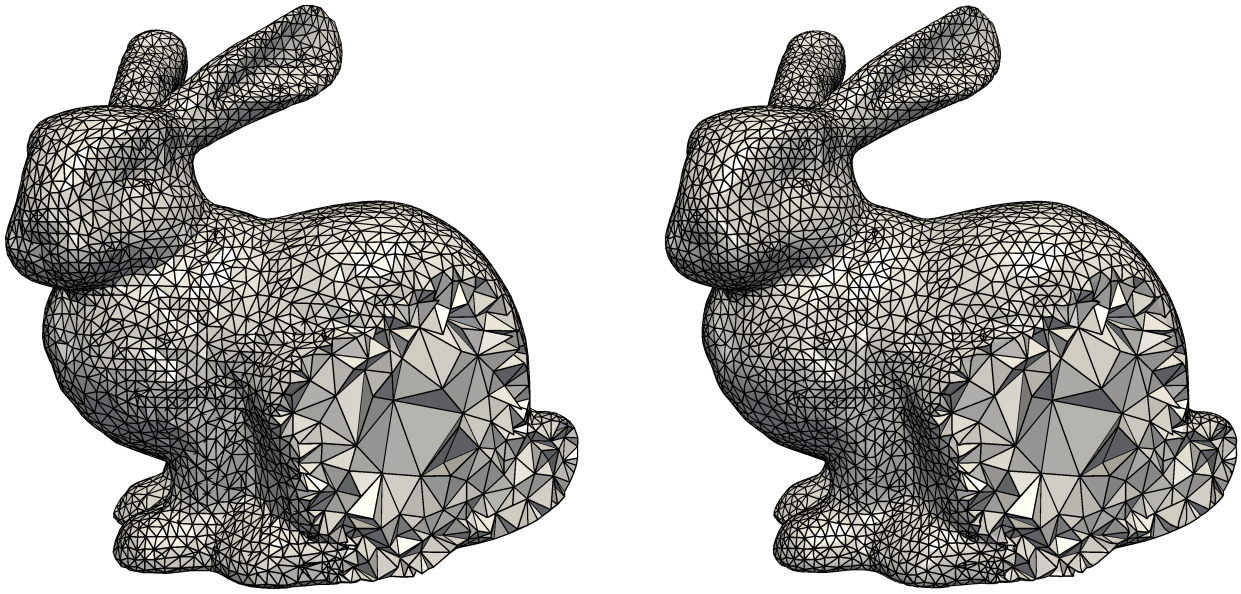
All the results have been obtained on a MacBook Pro with one dual-core Intel Core i7 CPU, with a clock frequency of 3.0 GHz, and total memory of 16 GBytes. The method presented in Section 4.2 is tested generating meshes of polynomial degree four.

5.1 The Standford bunny

A standard computer graphics 3D test model is the *Standford bunny*. The geometry composed of triangles is extracted from [Stanford University, 1993]. After the cleaning procedure, the linear tetrahedral mesh we use as input data has 28174 elements and 7734 nodes, see Figure 5.1(a). By means of the method presented in Section 4.2, we generate a mesh of polynomial degree

four. This mesh is composed of the same number of elements and 349063 nodes.

The resulting mesh took 11.52 seconds to be generated and is illustrated in Figure 5.1(b). We can appreciate the curvature on the boundary elements, while the interior high-order tetrahedra have straight-sided edges. Since the mesh is $\mathcal{C}^1$ and $\mathcal{C}^2$ almost everywhere, we can observe a significant improvement of the smoothness of the surface mesh.

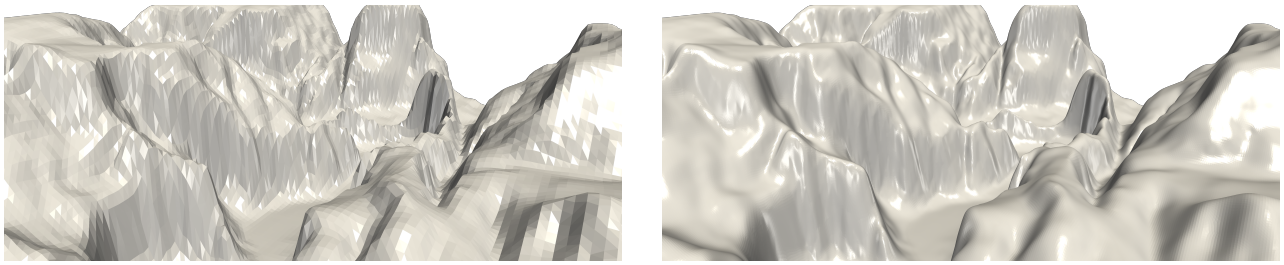


(a) **(b)**

Figure 5.1: Discretization of the Standford bunny. **(a)** Initial linear tetrahedral mesh. **(b)** Tetrahedral mesh of polynomial degree four.

5.2 Zion National Park

In this first topographical example, we consider a linear tetrahedral mesh of Zion National Park (Utah, USA). This geometry was extracted from [GrabCAD, 2009]. The mesh generated with *Tetgen*, Figure 5.2(a), has 294141 elements and 82994 nodes. The code took 125.18 seconds to generate the mesh of polynomial degree four of 3673943 nodes illustrated in Figure 5.2(b). All the vertices of the input mesh are regular nodes, so the high-order mesh generated by our method has as a boundary a $\mathcal{C}^2$-continuous surface. This is one of the features of our method when applied to topography scenarios. Most of these geometries are generated by means of measuring the height of a cartesian grid points distributed on the interested region. Thus, the triangle surface mesh is regular. This ensures that the mesh of polynomial degree four exactly matches the limit surface obtaining an exactly $\mathcal{C}^2$ mesh.

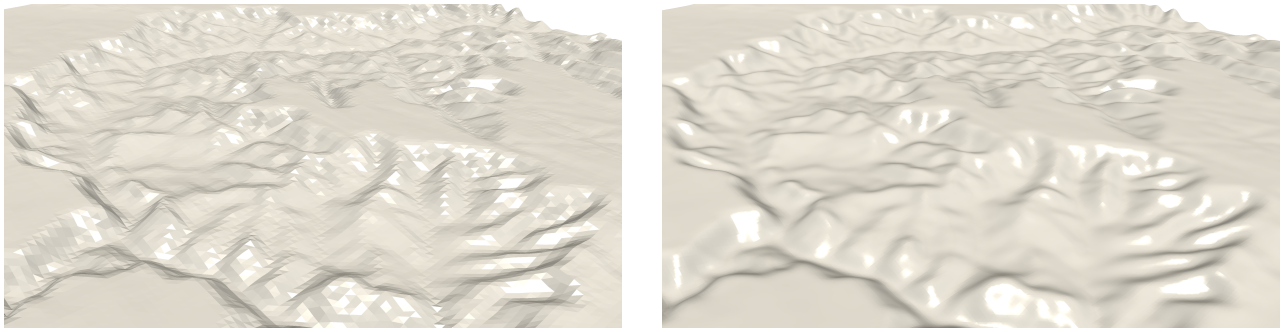


(a) **(b)**

Figure 5.2: Discretization of the Zion National Park with a **(a)** linear mesh and **(b)** the generated mesh of polynomial degree four.

5.3 Grand Canyon National Park

In this example, we consider a structured mesh of Grand Canyon National Park (Arizona, USA) obtained from [GrabCAD, 2009]. The mesh generated by *Tetgen* is composed of 386417 linear tetrahedra and 87591 nodes, see Figure 5.3(a). Hence, we generate a mesh of polynomial degree four with 386417 high-order tetrahedra and 4535303 nodes, as shown in Figure 5.3(b). This mesh takes 130.66 seconds to be generated. In this case, the original linear mesh is a structured mesh, i.e., each vertex is a regular node. Therefore, similarly to the previous case, the boundary of the high-order mesh is a $\mathcal{C}^2$-continuous surface.



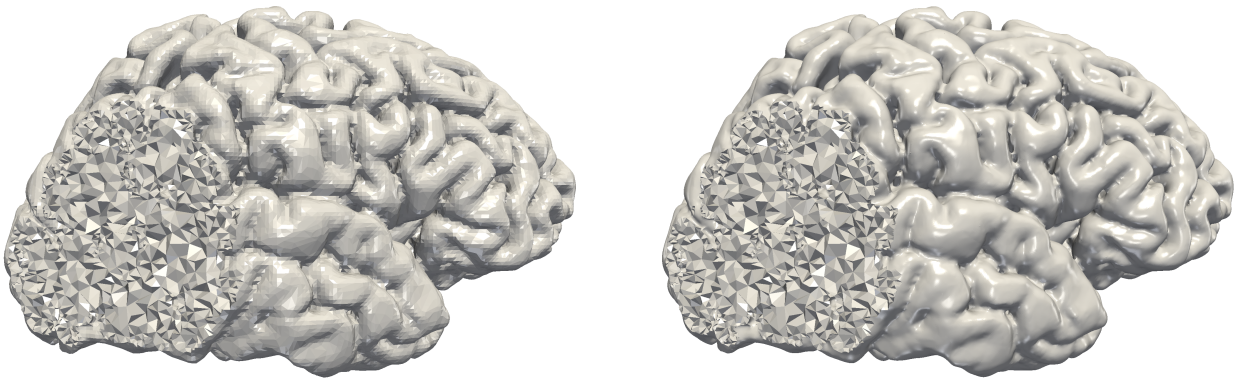
(a) **(b)**

Figure 5.3: Discretization of the Grand Canyon National Park with a **(a)** linear mesh and **(b)** the generated mesh of polynomial degree four.

5.4 Right hemisphere of the human brain

In this section, we test our method with a real case mesh obtained from tomography sectioning of the human brain obtained from [GrabCAD, 2009]. The initial linear tetrahedral mesh is composed of 514540 elements and 119113 nodes, see Figure 5.4(a). The mesh of polynomial

degree four is generated in 181.48 seconds and consists of 514540 elements and 6074369 nodes, as shown in Figure 5.4(b).

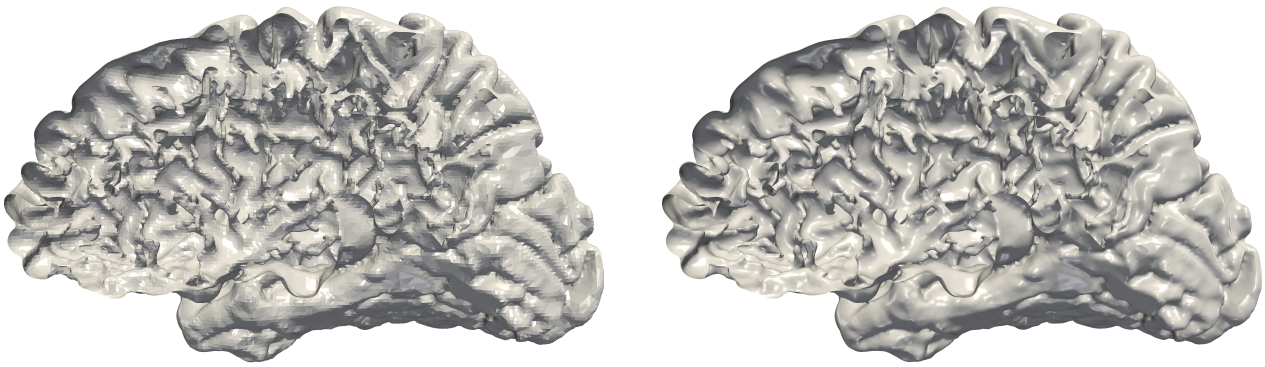


(a) **(b)**

Figure 5.4: Discretization of the right hemisphere of the human brain with a **(a)** linear mesh and **(b)** the generated mesh of polynomial degree four.

In this example, due to the achieved resolution on the data extraction procedure, we can observe clear artificial discontinuities of the normal of the boundary faces in the geometry, Figure 5.4(a). Our method is capable of obtaining a $\mathcal{C}^1$ surface mesh almost $\mathcal{C}^2$ everywhere. In the future, we would like to study the deviation of the high-order mesh from the original mesh.

This particular case is specially interesting due to the fact that, together with the outer boundaries, there are inner boundaries specially affected by the lack of resolution of the extraction procedure. After out mesh curving procedure these inner boundaries are also curved and the discontinuities of the normal of the triangle faces are clearly smoothed, see Figure 5.5. Once again, a further study of the deviation from the initial data set of the high order mesh would be of interest to analyze the effect of the mesh curving process.



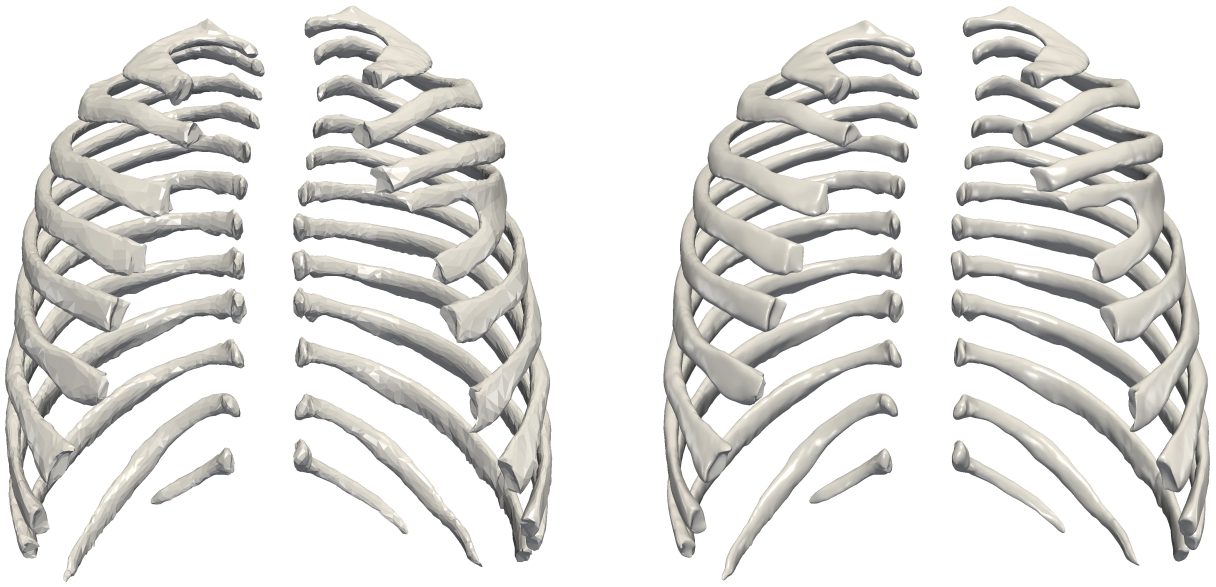
(a) **(b)**

Figure 5.5: Interior section of the right hemisphere of the brain. **(a)** Linear tetrahedral mesh. **(b)** Tetrahedral mesh of polynomial degree four.

5.5 Human rib cage

Finally, we consider a linear tetrahedral mesh with 129967 elements and 38969 nodes of a human rib cage extracted from [GrabCAD, 2009], see Figure 5.6(a). The high-order mesh is automatically generated in 62.68 seconds. It is composed of 1663101 nodes and the same number of elements but of polynomial degree four, see Figure 5.6(b).

Similarly to all the previous examples, we can observe the improvement on the discontinuities of the normal vectors of the boundary surface mesh. However, although we can assume that the original biological geometry is $\mathcal{C}^2$ and we obtain an almost $\mathcal{C}^2$ mesh, it is necessary to study in the future the deviation of the curved mesh from the sampled data. Although this data lacks the real smoothness of the actual ribs, the points that configure this surface geometry have been extracted from a tomography of the actual patient. Thus, measuring the deviation from this data would add significant value to the curving procedure.



(a) **(b)**

Figure 5.6: **(a)** Linear tetrahedral mesh of the human rib cage. **(b)** The generated high-order tetrahedral mesh.

Chapter 6

Conclusions and future research

In this work, two new methods to refine a linear tetrahedral mesh and to generate a high-order tetrahedral mesh have been presented. On the one hand, in Section 4.1 we proposed a new subdivision method for linear tetrahedral meshes when no additional information about the boundary is provided. The boundary of the resulting finer tetrahedral mesh tends to an almost everywhere $\mathcal{C}^2$-continuous surface. To do so, we subdivided each element into eight smaller tetrahedra. The boundary of the initial mesh is, in particular, a triangular surface mesh, so the Loop subdivision scheme detailed in Section 3.1 is applied. Finally, the boundary of the refined volume mesh is replaced by the finer surface mesh.

On the other hand, in Section 4.2 we presented a method to generate a high-order tetrahedral mesh from a linear tetrahedral mesh. For polynomial degree four, around regular points the boundary of the high-order mesh interpolates an almost everywhere $\mathcal{C}^2$-continuous surface; around extraordinary points this surface is approximated with fifth-order accuracy. We highlight that this method does not require an underlying curved geometry representation, only a linear tetrahedral mesh is used as input data. The method starts generating a straight-sided high-order mesh by increasing the polynomial degree of the physical elements. Next, from the boundary, a linear triangular surface mesh, we generate a high-order surface mesh by means of the method explained in Section 3.2. Finally, the boundary of the straight-sided tetrahedral mesh is replaced by the high-order surface mesh.

In the near future, we would like to propagate the curvature of the boundary to the interior nodes. This way, the interior nodes would be relocated to accommodate the curving and the mesh quality would improve. Furthermore, we would also like to compute the quality of the high-order mesh considering the original linear tetrahedron as the ideal element for each high-order element. Thus, the original mesh would have maximum quality and we would be able to compute the distortion of the generated mesh with respect to the given mesh. Together with an analysis of the validity and the quality of the mesh, in the near future we would like to measure the deviation of the curved meshes form the geometry data sets, to analyze the fidelity of the curved meshes to the triangle mesh geometry.

Alternatively, we would also like to define a functional quantifying the regularity of the mesh. Then, a smother boundary could be found solving an optimization problem subject to

some constraints.

The main limitation of the high-order mesh generation method we have proposed is that only polynomial degrees that correspond to powers of two can be chosen because each iteration of Loop's scheme subdivides each triangle into four smaller elements. Therefore, we would like to analyze other subdivision schemes, which may allow us more control on the degree of the high-order mesh. Nevertheless, Loop's scheme seems promising when moving to high-order mesh generation in higher-dimensions.

Bibliography

[Antepara et al., 2015] Antepara, O., Lehmkuhl, O., Borrell, R., Oliet, C., and Oliva, A. (2015). Parallel mesh multiplication and adaptation technique for turbulent flow simulation using unstructured meshes. In *International Conference on Parallel Computational Fluid Dynamics*.

[Babuška et al., 1981] Babuška, I., Szabó, B. A., and Katz, I. N. (1981). The p-version of the finite element method. *SIAM journal on numerical analysis*, 18(3):515–545.

[Bajaj et al., 2002] Bajaj, C., Schaefer, S., Warren, J., and Xu, G. (2002). A subdivision scheme for hexahedral meshes. *The Visual Computer*, 18(5):343–356.

[Borja, 2017] Borja, M. S. (2017). Generation of curved high-order meshes from linear meshes using the subdivision method.

[Catmull and Clark, 1978] Catmull, E. and Clark, J. (1978). Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350 – 355.

[Chaurasia et al., 2012] Chaurasia, H., Roca, X., Persson, P., and Peraire, J. (2012). A coarse-to-fine approach for efficient deformation of curved high-order meshes. *Research Notes, 21st Int. Meshing Roundtable, Springer International Publishing*, pages 1–5.

[Cignoni et al., 2008] Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. (2008). MeshLab: an Open-Source Mesh Processing Tool. In Scarano, V., Chiara, R. D., and Erra, U., editors, *Eurographics Italian Chapter Conference*. The Eurographics Association.

[Frey and George, 2000] Frey, P. J. and George, P.-L. (2000). *Mesh Generation Application to Finite Elements*. ISTE Publishing Company, illustrated edition edition.

[Gargallo-Peiró, 2014] Gargallo-Peiró, A. (2014). *Validation and generation of curved meshes for high-order unstructured methods*. PhD thesis, Universitat Politcnica de Catalunya. Departament de Matemtica Aplicada III.

[Gargallo-Peiró et al., 2017] Gargallo-Peiró, A., Houzeaux, G., and Roca, X. (2017). Subdividing triangular and quadrilateral meshes in parallel to approximate curved geometries. *Procedia Engineering*, 203:310 – 322. 26th International Meshing Roundtable, IMR26, 18-21 September 2017, Barcelona, Spain.

[Gargallo-Peiró et al., 2015a] Gargallo-Peiró, A., Roca, X., Peraire, J., and Sarrate, J. (2015a). Distortion and quality measures for validating and generating high-order tetrahedral meshes. *Engineering with Computers*, 31(3):423–437.

[Gargallo-Peiró et al., 2015b] Gargallo-Peiró, A., Roca, X., Peraire, J., and Sarrate, J. (2015b). Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, 103(5):342–363.

[Gargallo-Peiró et al., 2013] Gargallo-Peiró, A., Roca, X., Sarrate, J., and Peraire, J. (2013). Inserting curved boundary layers for viscous flow simulation with high-order tetrahedra. *22nd International Meshing Roundtable, Orlando, Florida*.

[GrabCAD, 2009] GrabCAD (2009). GrabCAD. `https://grabcad.com/`. [Online; accessed August-2018].

[Greiner and Grosso, 2000] Greiner, G. and Grosso, R. (2000). Hierarchical tetrahedral-octahedral subdivision for volume visualization. *The Visual Computer*, 16(6):357–369.

[Hesthaven and Warburton, 2007] Hesthaven, J. S. and Warburton, T. (2007). *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media.

[Hughes, 1987] Hughes, T. J. R. (1987). *The finite element method: linear static and dynamic finite element analysis*. Prentice-Hall.

[Jiao and Wang, 2012] Jiao, X. and Wang, D. (2012). Reconstructing high-order surfaces for meshing. *Engineering with Computers*, 28(4):361–373.

[Johnen et al., 2013] Johnen, A., Remacle, J.-F., and Geuzaine, C. (2013). Geometrical validity of curvilinear finite elements. *Journal of Computational Physics*, 233:359–372.

[Jones et al., 2001] Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. [Online; last accessed September-2018].

[Klöckner et al., 2013] Klöckner, A., Warburton, T., Hesthaven, J., and Zhu, X. (2012-2013). modepy. [Online; last accessed March-2018].

[Kobbelt, 2000] Kobbelt, L. (2000). Sqrt(3)-subdivision. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*.

[Loop, 1987] Loop, C. (1987). *Smooth Subdivision Surfaces Based on Triangles*. PhD thesis, Department of Mathematics, The University of Utah, Masters Thesis.

[Loop, 2002] Loop, C. (2002). Bounded curvature triangle mesh subdivision with the convex hull property. *The Visual Computer*, 18(5):316–325.

[Moxey et al., 2016] Moxey, D., Ekelschot, D., Keskin, ., Sherwin, S., and Peir, J. (2016). High-order curvilinear meshing using a thermo-elastic analogy. *Computer-Aided Design*, 72:130 – 139. 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[Moxey et al., 2015a] Moxey, D., Green, M., Sherwin, S., and Peiró, J. (2015a). An isoparametric approach to high-order curvilinear boundary-layer meshing. *Computer Methods in Applied Mechanics and Engineering*, 283:636–650.

[Moxey et al., 2015b] Moxey, D., Hazan, M., Sherwin, S., and Peiro, J. (2015b). Curvilinear mesh generation for boundary layer problems. In *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach*, pages 41–64. Springer.

[Persson et al., 2006] Persson, P.-O., Aftosmis, M. J., and Haimes, R. (2006). On the use of loop subdivision surfaces for surrogate geometry. In Pébay, P. P., editor, *Proceedings of the 15th International Meshing Roundtable*, pages 375–392, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Persson and Peraire, 2008] Persson, P.-O. and Peraire, J. (2008). Curved mesh generation and mesh refinement using lagrangian solid mechanics.

[Quarteroni and Quarteroni, 2009] Quarteroni, A. and Quarteroni, S. (2009). *Numerical models for differential problems*, volume 2. Springer.

[Ruiz-Gironés et al., 2016a] Ruiz-Gironés, E., Roca, X., and Sarrate, J. (2016a). High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3d cad representation. *Computer-Aided Design*, 72:52–64.

[Ruiz-Gironés et al., 2016b] Ruiz-Gironés, E., Sarrate, J., and Roca, X. (2016b). Generation of curved high-order meshes with optimal quality and geometric accuracy. *Procedia engineering*, 163:315–327.

[Si, 2015] Si, H. (2015). Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software (TOMS)*, 41(2):11.

[Stam, 1998] Stam, J. (1998). Evaluation of loop subdivision surfaces.

[Stanford University, 1993] Stanford University, C. G. L. (1993). Stanford Bunny. `http://graphics.stanford.edu/data/3Dscanrep/`. [Online; accessed August-2018].

[Staten et al., 2016] Staten, M., Carnes, B., Mcbride, C., Stimpson, C., and Cox, J. (2016). Mesh scaling for affordable solution verification. *Procedia Engineering*, 163:46 – 58. 25th International Meshing Roundtable.

[Szabó and Babuška, 1991] Szabó, B. A. and Babuška, I. (1991). *Finite element analysis*. John Wiley & Sons.

[Van Rossum and Drake Jr, 1995] Van Rossum, G. and Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

[Xie et al., 2013] Xie, Z. Q., Sevilla, R., Hassan, O., and Morgan, K. (2013). The generation of arbitrary order curved meshes for 3d finite element analysis. *Computational Mechanics*, 51(3):361–374.

[Yilmaz and Aliabadi, 2011] Yilmaz, E. and Aliabadi, S. (2011). Mesh multiplication technique with surface correction. *PARCFD2011, Barcelona (Spain)*.

[Yilmaz and Aliabadi, 2013] Yilmaz, E. and Aliabadi, S. (2013). Surface conformed linear mesh and data subdivision technique for large-scale flow simulation and visualization in variable intensity computational environment. 80:388402.